Lecture 3

Karnaugh map



Karnaugh map

- Due to the difficulty of using Boolean algebra in simplifying the Boolean function, we need to seek another less complicated method for simplification.
- Another problem associated with using Boolean algebra is that we can not tell for sure if the form of the function we have reached after many steps of simplification is the minimum solution for that function or not.
- We will study another faster and easier method for simplification called the Karnaugh map.



Karnaugh map

- Historically speaking, it was first proposed by Vietch and modified by Karnaugh.
- This map provides a systematic procedure for simplifying up to 5variables.
- Karnaugh map for more than 5 variables are difficult to handle.
- However, other methods such as Quine-MacCliskey algorithm can be used to simplify higher number of variables functions.



What is Karnaugh map?

- Karnaugh map is made of squares, where each square represents one minterm.
- Thus the number of squares in the map is equal to the number of possible number of minterms for a given number of variables.
- For example, if the number of variables =3, then the map will contain 8 squares, while if the number of variables =4, the map will contain 16 squares, etc…
- The squares are organized into rows and columns.
- Above each column and beside each row variables values are written such that each square is read in terms of row and column intersection.
- That is why we can say that the map is a visual method of representation a truth table.



2-variable Karnaugh map

- A 2- variable Karnaugh map is shown in the figure.
- As we have 2 variables which produces 4 possible minterms, Karnaugh map contains 4 squares each for one minterm.
- The binary values of the variables are distributed above each column and beside each row such that each intersection between a column and a row represents a minterm.
- Binary values of variable A are the above the columns while binary values of variable B are beside the rows.

B \ A	0	1
0	m_0	m_2
1	m_1	m_3

B \ A	0	1
0	00	10
1	01	11



Representing a Boolean function in the map

- Like truth table where we put '1' in the function column in the place of combinations it equals '1' and put '0' otherwise, we put '1' in the squares where the function equals '1' and put '0' otherwise.
- For example consider the truth table below.

Inputs		Output
A	B	C
0	0	1
0	1	0
1	0	1
1	1	0

A / B	0	1
0	1	1
1		



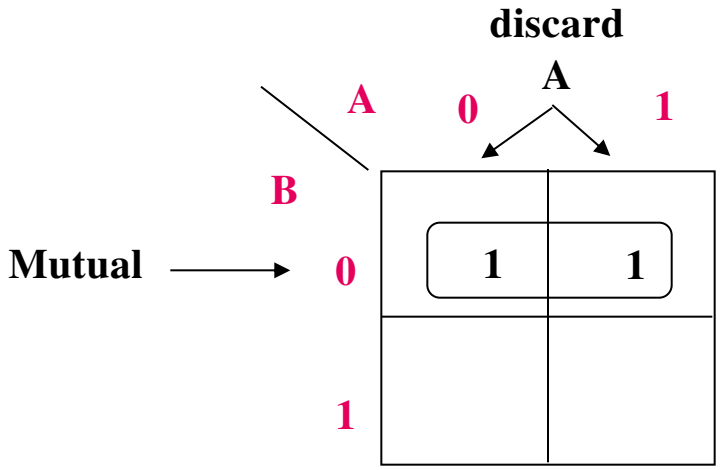
Using Karnaugh map to simplify a Boolean function

- Before describing how to use the map in simplification, let's have a close look at the map in the previous example.
- We find that there are **two adjacent** squares having the value '1' (00 and 10) which correspond to **(A/B/ , AB/)**.
- Both terms have **B/** and vary in **A** and **A/** . They can be simplified into:

 A/B/ + AB/ = B/ (A/+A) = B/

- From this we can say that if there are adjacent squares, we can deduce that they have mutual variables and different variables.
- We, as in the example, can use these squares to discard at least one of the variables which are different yielding the mutual variable(s).





- In the example, we discarded the variable A as it was different in the two squares, yielding the mutual variable **B/**.
- ***Thus as a general rule, if we combined two squares, we can discard one variable, if we combined 4 squares, we can discard 2 variables and so on.***
- To indicate that we will use these squares for simplification, we put a rounded square around these '1'.

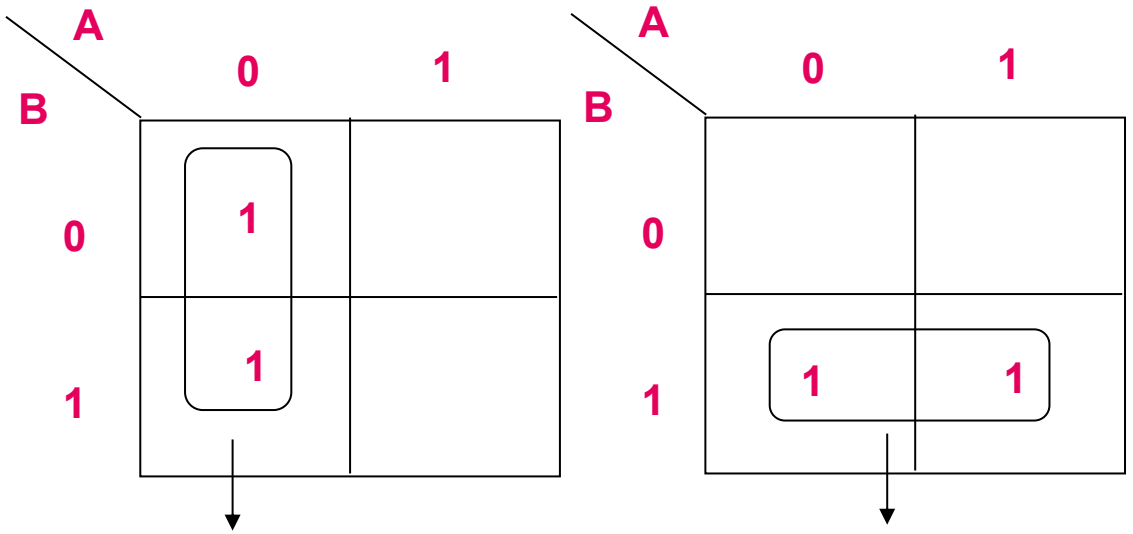


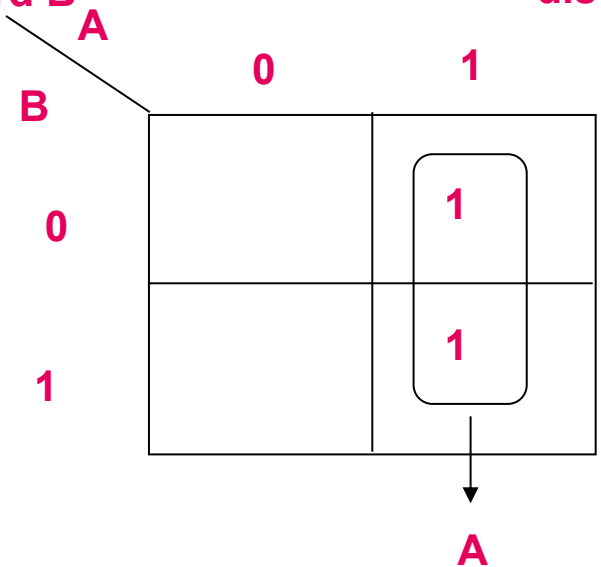
- **Drill:**

 Using 2-variable Karnaugh map prove that combining 4 squares will discard two variables.



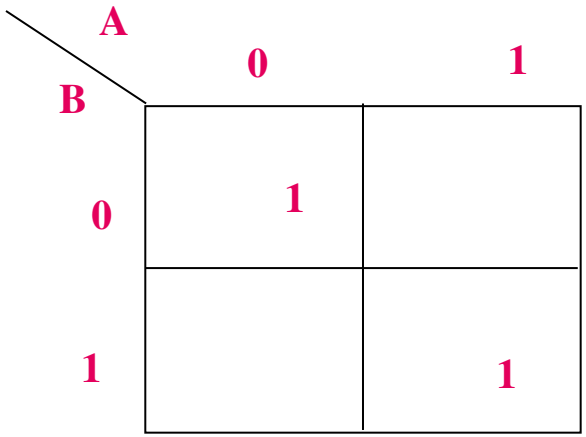
- Other possible cases of adjacent squares containing '1' and the result of combining them are also shown.







- But the non adjacent squares can not be combined as they can not be simplified.
- Consider the case shown below, the first square corresponds to (**A/B/** , **AB**). These two terms can not be further simplified as they do not have a mutual variable.



Drill

What is the other case of '1"s positions in the map where we can not join them for simplificatio



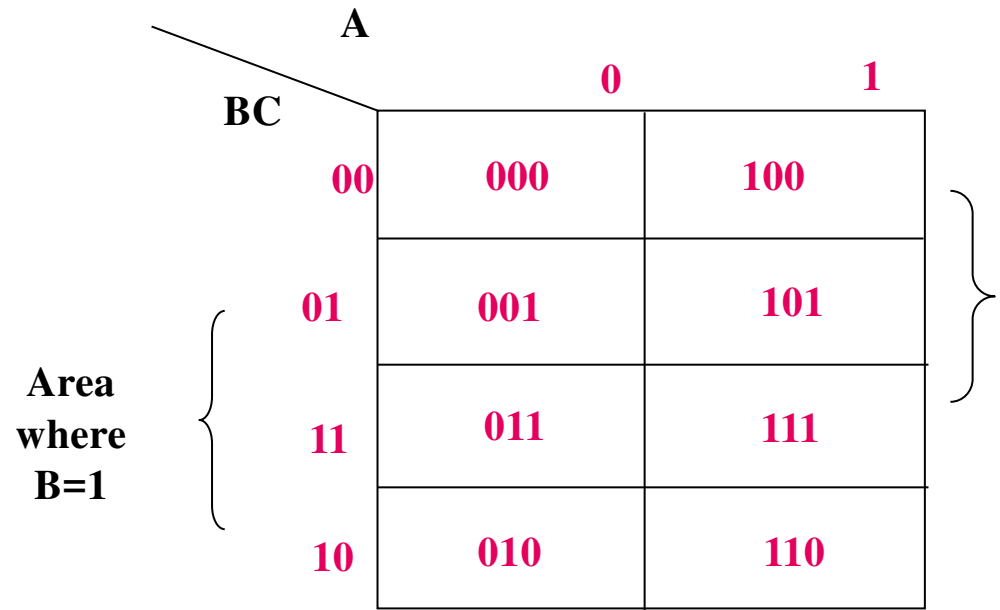
3-variable Karnaugh map

- As we have 3 variables which produces 8 possible minterms, Karnaugh map contains 8 squares each for one minterm.
- The binary values of the variables are distributed above each column and beside each row such that each intersection between a column and a row represents a minterm.
- To read the minterm in a specific square, make an intersection between a column and a row.

BC \ A	0	1
00	m_0	m_4
01	m_1	m_5
11	m_3	m_7
10	m_2	m_6



- *For example, intersection between first column and first row produces (000) which is minterm m0, and intersection between first column and second row produces (001) which is minterm m1, etc…*
- *The map can be viewed as areas where each area contains 4 squares that contains a specific variable = '1'.*
- *Thus, in case of 3-variabel map, we have 3 areas one for each variable as shown in the figure. It will be shown later in this chapter how to use these areas.*





Representing a Boolean function in the map

- As discussed before in 2-variable map, we put '1' in the square where the function = '1' as shown in the given example below.

Inputs			Output
A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

A / BC	0	1
00		1
01		1
11	1	1
10		1



Using Karnaugh map to simplify a Boolean function

- From the map, we find that there are four adjacent squares having the value '1' (100, 101, 111 and 110) which correspond to (AB/C/, AB/C, ABC and ABC/).

BC \ A	0	1
00		1
01		1
11	1	1
10		1

- The four terms have a mutual variable A and vary in other two variables.
- They can be simplified into:

 AB/C/+AB/C+ABC+ABC/

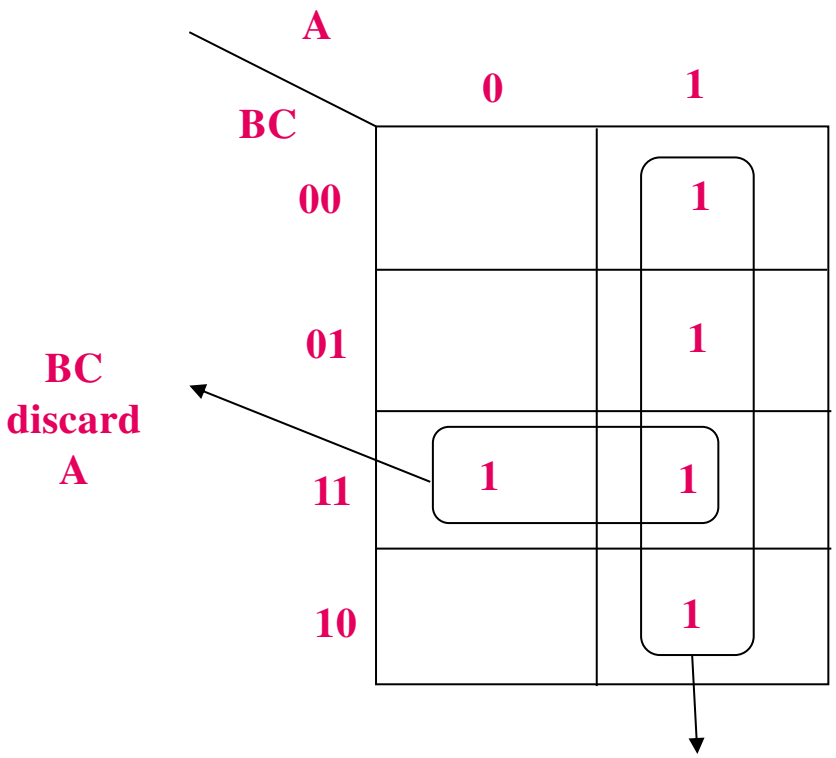
 = AB/(C/+ C)+AB(C+C/)

 = AB/ +AB = A(B/ +B) = A
- Combining the four adjacent squares, discarded the two variables B and C as they were different in the four squares, yielding the mutual variable A.



- Thus the function after simplification is:

$$F = A + BC$$





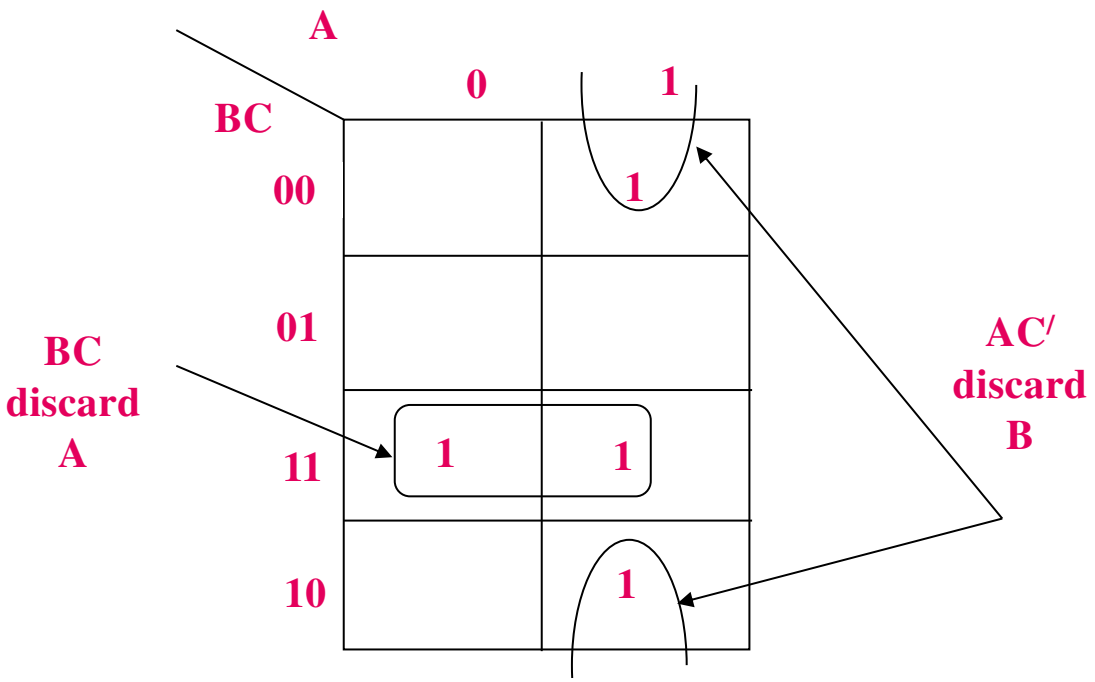
- **Drill:**

Using 3-variable Karnaugh map simplify the following function.

Inputs			Output
A	B	C	W
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



Thus, we can combine the two squares as shown in figure as they have (**AC/**) mutual and B different so it id discarded.





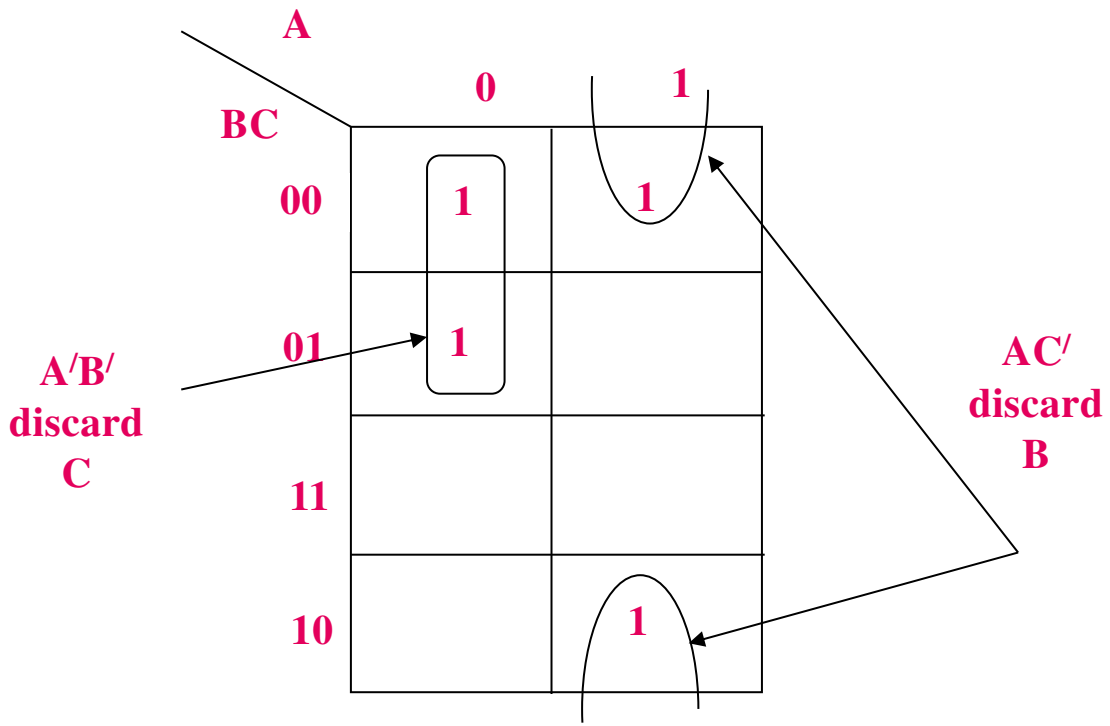
Representing the sum of minterms form in the map

- It is very common to have the function represented as a sum of minterm, so all what we have to do to put it in the map is to locate the place of minterm (from the above minterm location in the map) and put '1' in these squares.
- For example, we have the following function represented in sum of minterm form:

$$F(A,B,C) = \sum m(0, 1, 4, 6)$$

- Knowing the minterms positions in the map, we can put the '1's in the proper place as follows.





$$F(A,B,C) = \sum m(0, 1, 4, 6)$$

Thus the simplified function is:

F = A/B/ + AC/



Drill

Using 3-variable Karnaugh map simplify the following function.

$$F(A,B,C) = \sum m(0, 1, 4, 6)$$



Representing the product of maxterms form in the map

- We can also represent the function if it is represented in product of maxterm by converting it into sum of minterms.
- For example, we have the following function represented in sum of minterm form:

F (A,B,C) = Π M(1, 2, 3, 6)

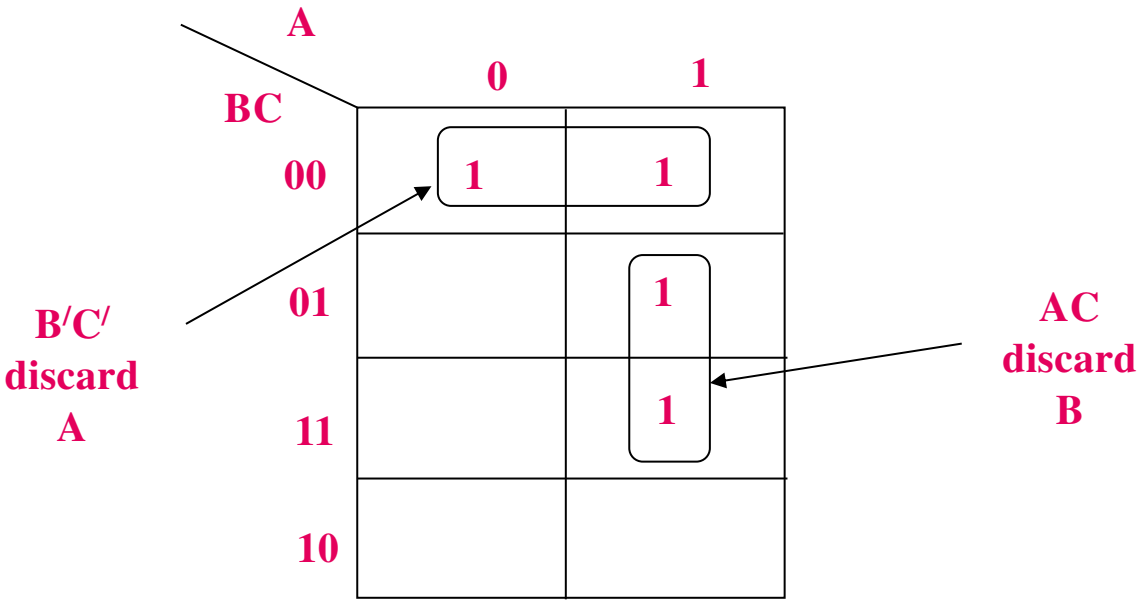
- First let's convert it into sum of product form (by taking the not available terms in this form as we discussed previously) to be:

F (A,B,C) = ∑ m(0, 4, 5, 7)



F (A,B,C) = ∑ m(0, 4, 5, 7)

- Knowing the minterms positions in the map, we can put the '1's in the proper place as follows:



- Thus the simplified function is:

F = B/C/ + AC



- **Drill**

 Using 3-variable Karnaugh map simplify the following function.

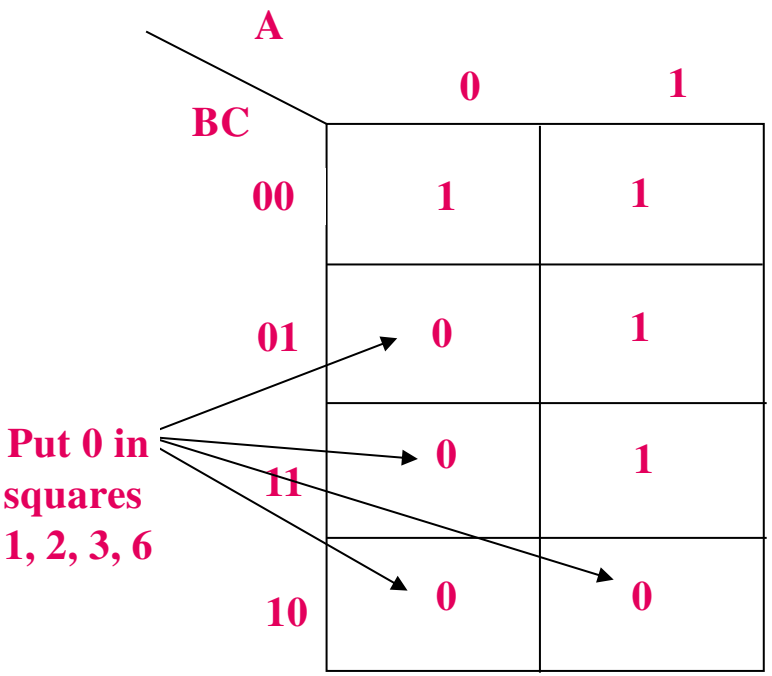
 $F(A,B,C) = \Pi M(0, 1, 4, 6)$



- Another way to represent the product of maxterms on the map is to put '0' in the square assuming that the given maxterms is at the same place of minterms then complete the rest of the squares with '1"s as follows. For the same function:

$$F(A,B,C) = \Pi\ M(1, 2, 3, 6)$$

- We will put '0' in squares corresponding to minterms (1, 2, 3, 6).





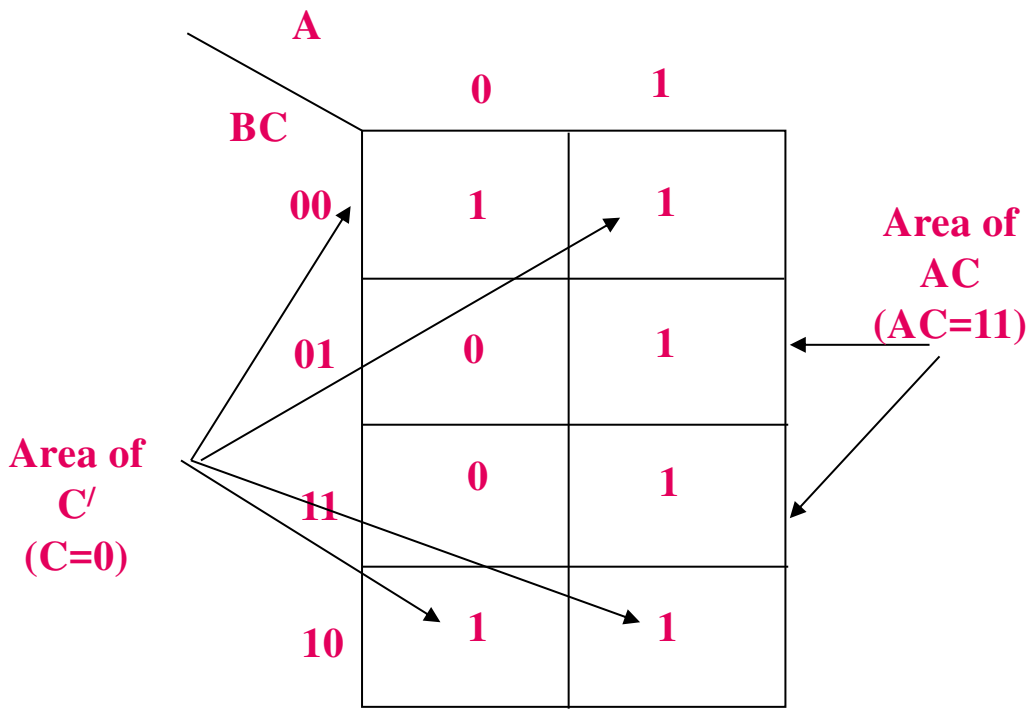
Representing the sum of products form in the map

- Sum of product is not the formal form of representing the function, while each square corresponds to a minterm.
- However, we necessary do not have to convert it into sum of minterm to be able to represent it in the map.
- We can make use of knowing the variables area denoted in previous figure to locate the squares that will carry '1'.
- For example, if the function represented on the form of sum of products as follows:

 F = C/ + AC

- We will locate the area of C then its opposite will be C/.
- For AC we will have to locate the area where AC=11 as shown below.





F = C/ + AC

- **Notice that if the term has two variables (from the three variables), we must put two '1"s, while if the term has one variable (from the three variables), we must put four '1's (the opposite of combining two squares will omit one variable and 4 squares will omit 2 variables).**



Drill

Using 3-variable Karnaugh map represent the following function.

$$F(A,B,C) = AB/ + C$$



4-variable Karnaugh map

- A 4-variable Karnaugh map is shown in the figure below.
- It contains 16 squares each for one minterm.

CD \ AB	00	01	11	10
00	m_0	m_4	m_{12}	m_8
01	m_1	m_5	m_{13}	m_9
11	m_3	m_7	m_{15}	m_{11}
10	m_2	m_6	m_{14}	m_{10}



Area where A=1

CD \ AB	00	01	11	10
00	0000	0100	1100	1000
01	0001	0101	1101	1001
11	0011	0111	1111	1011
10	0010	0110	1110	1010

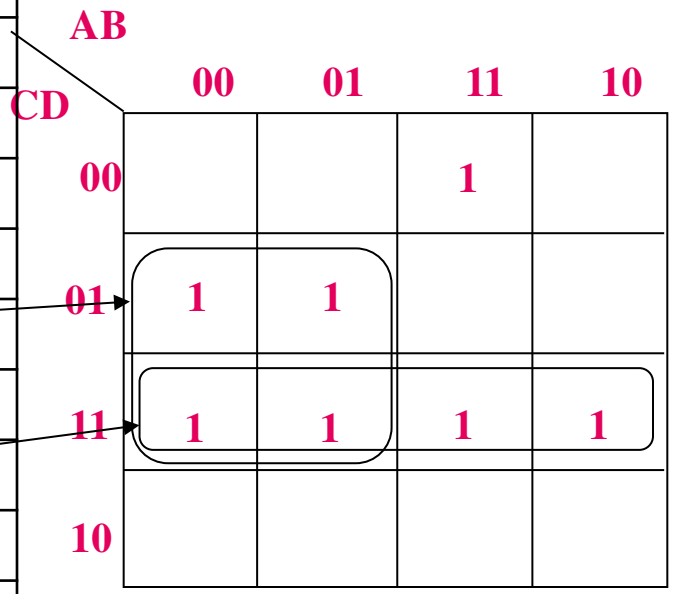
Area where C=1

Area where D=1

Area where B=1



Inputs				Output
A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1



F = A/D +CD



Drill

Using 3-variable Karnaugh map simplify the following function.

Inputs				Output
A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

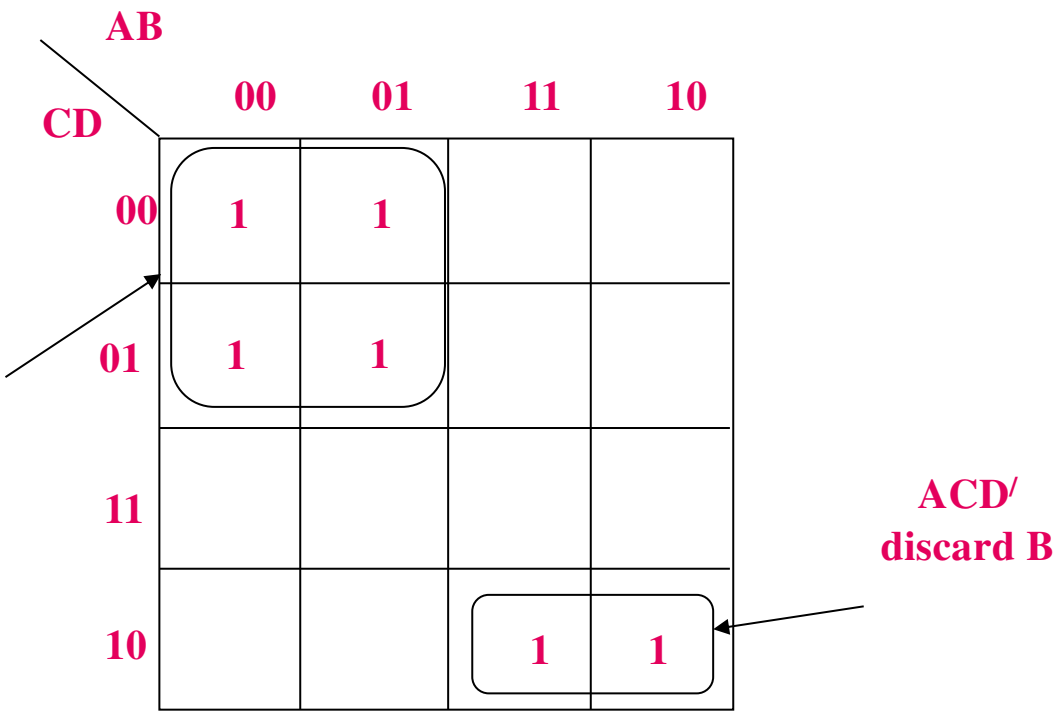


If we have the following function represented as sum of minterms, we can simplify it as follows:

F (A,B,C,D) = ∑ m(0, 1, 5, 6, 10, 14)

Thus the function after simplification is:

F (A,B,C,D) = A/C/ + ACD/





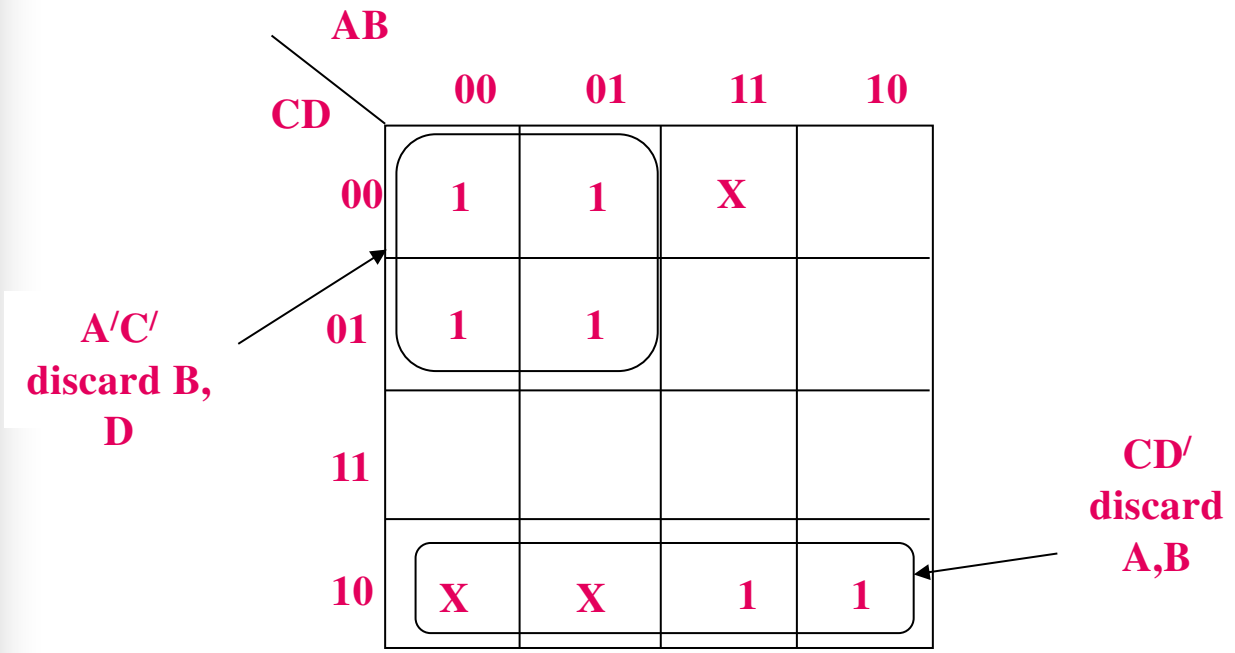
- **Drill**
- **Using 3-variable Karnaugh map simplify the following function. F (A,B,C,D) = ∑ m(1, 4, 6, 11, 12, 15)**



Simplification with don't care conditions

- Don't care conditions are represented in the map as X's.
- These X's are neutral, i.e. sometimes it is treated as '1' and others are treated as '0' according to the situation. I
- f it can be used to further simplify a number of squares, it is treated as '1'.
- While if it will not help in the simplification process, it will be treated as '0'.
- In the next example, we will see both cases of treating don't care squares while we are simplifying a function.





- We have three don't cares as shown in the map, we used two of them with two '1' squares to discard variable (**A**), while the upper one could not be used as it will not do any good.



- ***Drill***
- ***Using 3-variable Karnaugh map simplify the following function.***

 F (A,B,C,D) = ∑ m(1, 4, 6, 11, 12, 15)

 Don't care = ∑ m(0, 5, 10, 13)



Simplifying the Boolean function into product of sums using the map

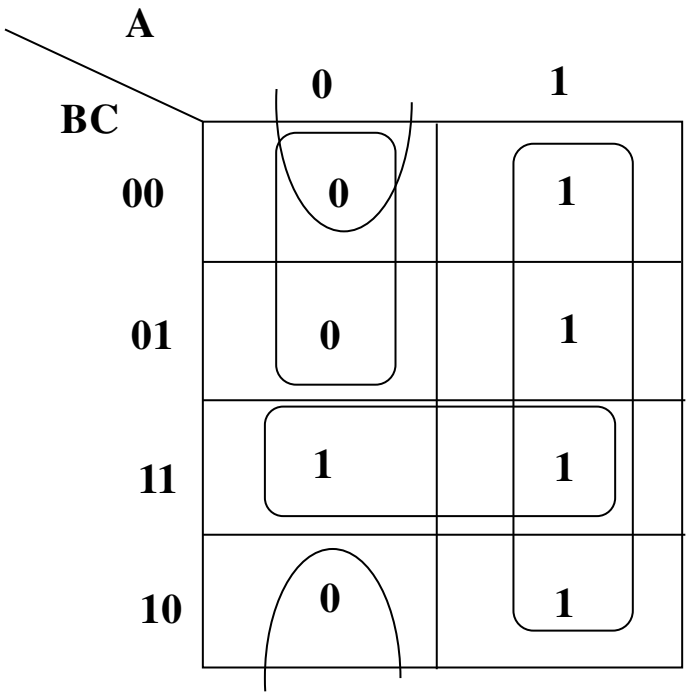
- Sometimes, due to the implementation constrains, it is desired to implement the function as product of sums using a bank of OR gates and an AND gate.
- To obtain the function in the form of product of sums, rather the sum of products form we used to have from the map, we can:

1. instead of simplifying the '1' squares to obtain the function **F**, we will simplify the '0' squares to obtain the inverse of the function **F/** in the form of sum of product.

2. Take the complement of **F/** to obtain **F** again which will produce the desired product of sums form.



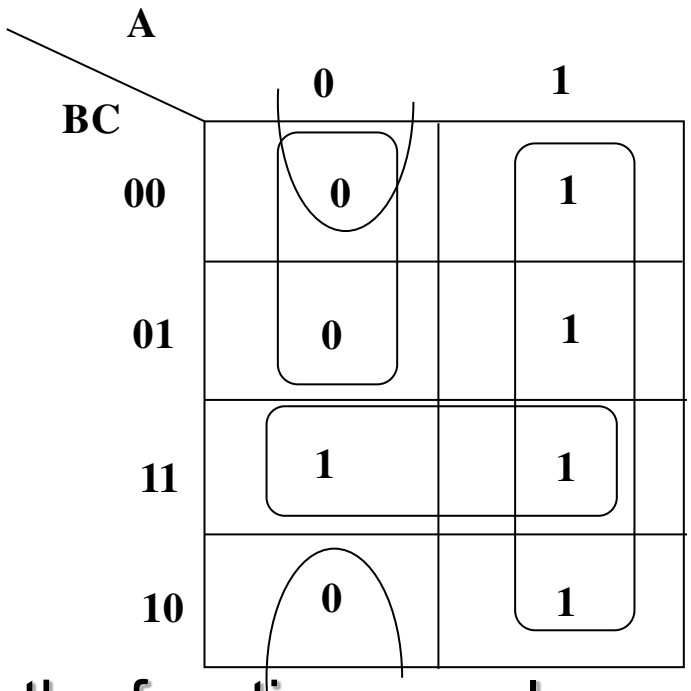
- **For the following function represented in the map, we can get both forms product of sums and sum of products**



Thus the simplified function as sum of products is (was we seen before):

F = A + BC





- While the function complement expressed as sum of products is:

 F/ = A/B/ + A/C/

- Taking the complement F/ of we get F in the form of product of sums:

 (F/)/ = (A/B/ + A/C/)/

 F = (A/B/)/ (A/C/)/

 = (A +B) (A+C)



Drill

Using 3-variable Karnaugh map simplify the following function as product of sums.

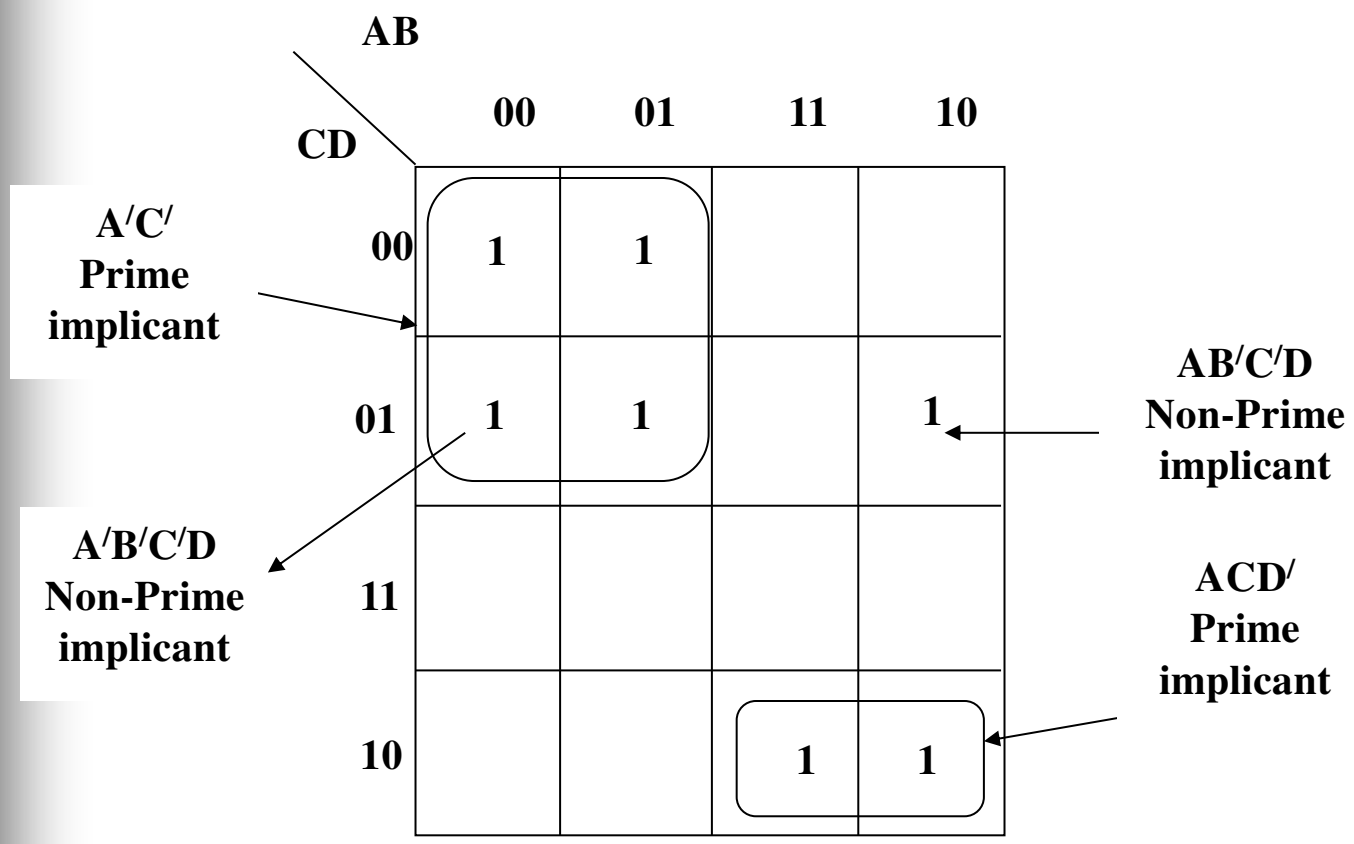
$F(A,B,C,D) = \sum m(1, 4, 6, 11, 12, 15)$



Determination of minimum expressions using essential prime implicants

- In a Karnaugh map, any single '1' square or any group of '1"s squares, which can be combined together to remove variable(s), represented as a product term is called an implicant of the function F.
- There are two types of implicants: prime and non-prime.
- A product term is called a prime implicant if it can not be further combined with other terms to eliminate variable (s).
- While, a non-prime implicant term is the term which can be combined with other terms to eliminate variable(s).




AB
CD
00
01
11
10
00
01
11
10
1
1
1
1
1
1
1
1
A′C′
Prime
implicant
A′B′C′D
Non-Prime
implicant
AB′C′D
Non-Prime
implicant
ACD′
Prime
implicant



- The minimum sum of products form for a function consists of some (but not necessary all) of the prime implicants.
- In other words, if the sum of products form of a function contains a term which is not a prime implicant, it is not called minimum form.
- This implies that to get the minimum sum of products form, we have to find the minimum number of prime implicant terms that covers all of the 1's on the map.



AB \ CD	00	01	11	10
00		1	1	
01	1	1	1	
11	1		1	1
10			1	1

- In the above map, we have 10 squares having '1'.
- There are six prime implicants:
- A/B/D/, BC/, AC, A/C/D, AB and B/CD.
- The first three implicants covers all the 10 1's, thus they are sufficient for the simplification and they constitute the minimum solution.
- The last three implicants are not part of the minimum solution and will not be used.

F = A/B/D/ + BC/ + AC